8. Feature Extraction from Images

Aim of this Chapter: Learn the Basic Feature Extraction Methods for Images

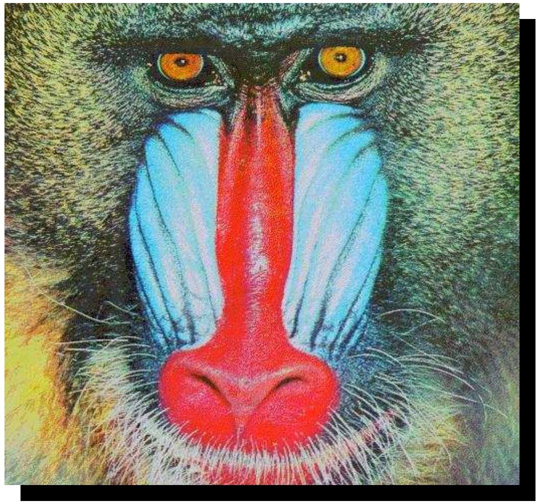
Main features:

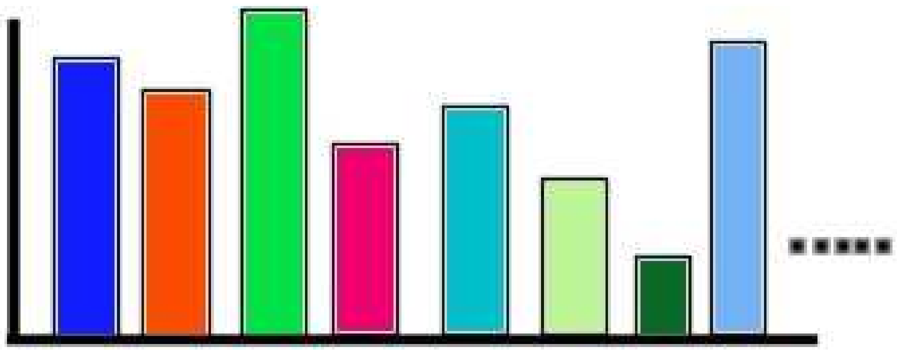
- Color
- Texture
- Edges

8.1. Histograms and Color Features

Color Histogram

Calculate percentage of color present in image





Deficiency: loss of regional information

Measurements at Pixels

- An image I is a set of pixels
- At each pixel:
  - Measure some m-dimensional property

Example:

Each pixel of an RGB image is a 3-dimensional vector

Formally:

$$f_I : R \subset \mathbb{R}^2 \rightarrow M \subset \mathbb{R}^m$$

Create a finite Partition of M

Create finite partition of M:

$$M = \bigcup_{k=1}^{K} B_k$$

B_k are subsets of M

B_k are called bins and k is the label of the bin

Example

Example:

Let M be the grey levels of an image $M = [0:255]$

Label of bin	Gray levels B_k
1	0-31
2	32-63
3	64-95
4	96-127
5	128-159
6	160-191
7	192-213
8	214-255

From Bins to Histograms

Indicator function:

$$b_k(x) = \begin{cases} 1 & \text{if } f_I(x) \in B_k \\ 0 & \text{otherwise} \end{cases}$$

x is element of the image

From Bins to Histograms

A histogram is a vector

$$\vec{\mathrm{H}} = (h_1, \ldots h_K)$$

with

$$h_k = \frac{\int\limits_{x \in R} b_k(f_I(x))dx}{\int\limits_{x \in R} 1dx}$$

Histogram Distances

Motivation:

measures the similarity of

- images
- speech
- music

Issue:

how to capture perceptual similarity

Histogram Distances

L_1 distance (Manhattan distance)

$$d_1(H,L) = \sum_{k=1}^{K} | h_k - l_k |$$

L_2 distance (euklidian distance)

$$d_2(H,L) = \sqrt{\sum_{k=1}^{K} | h_k - l_k |^2}$$

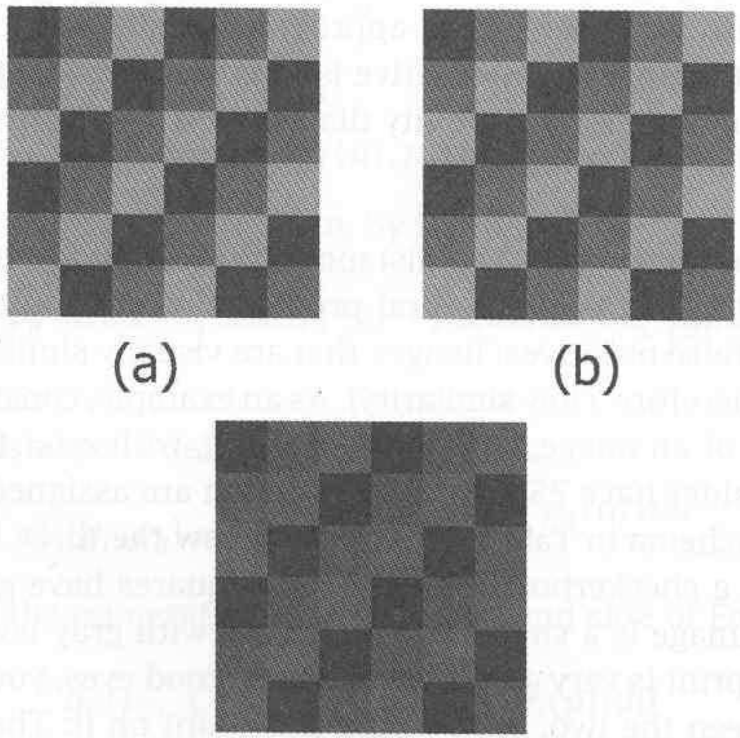
L_∞ distance (maximum distance)

$$d_\infty(H,L) = \max_k(| h_k - l_k |)$$

Exercise

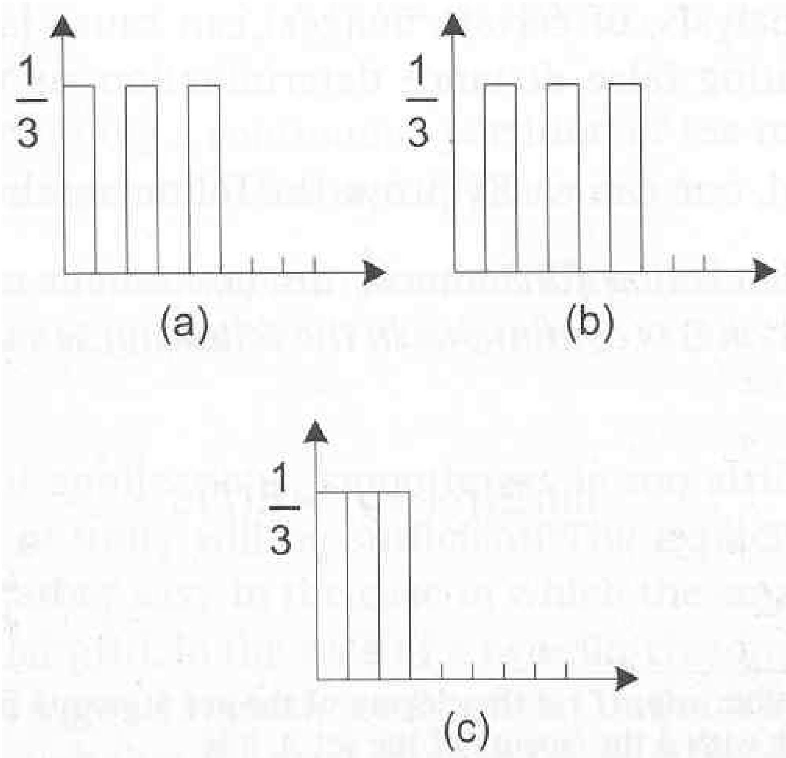
Microsoft
werPoint-Präsentati

Example for potential problem with histogram distance



(a) (b) (c)

Example for potential problem with histogram distance

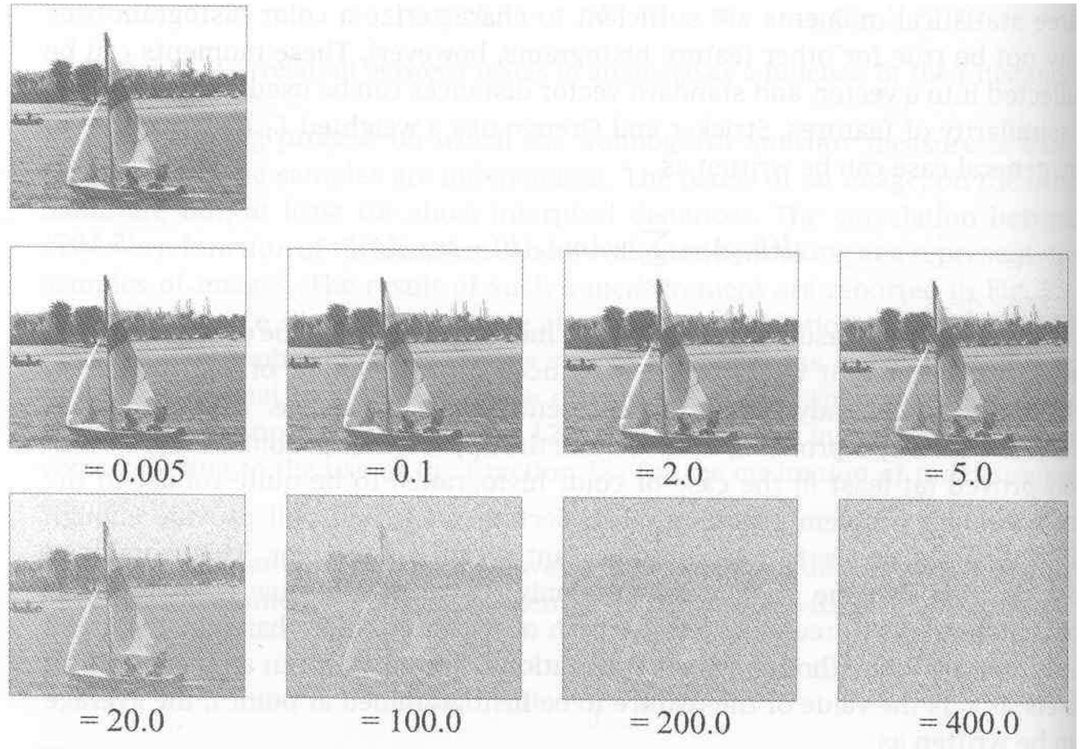


Distances of the three checkerboard images

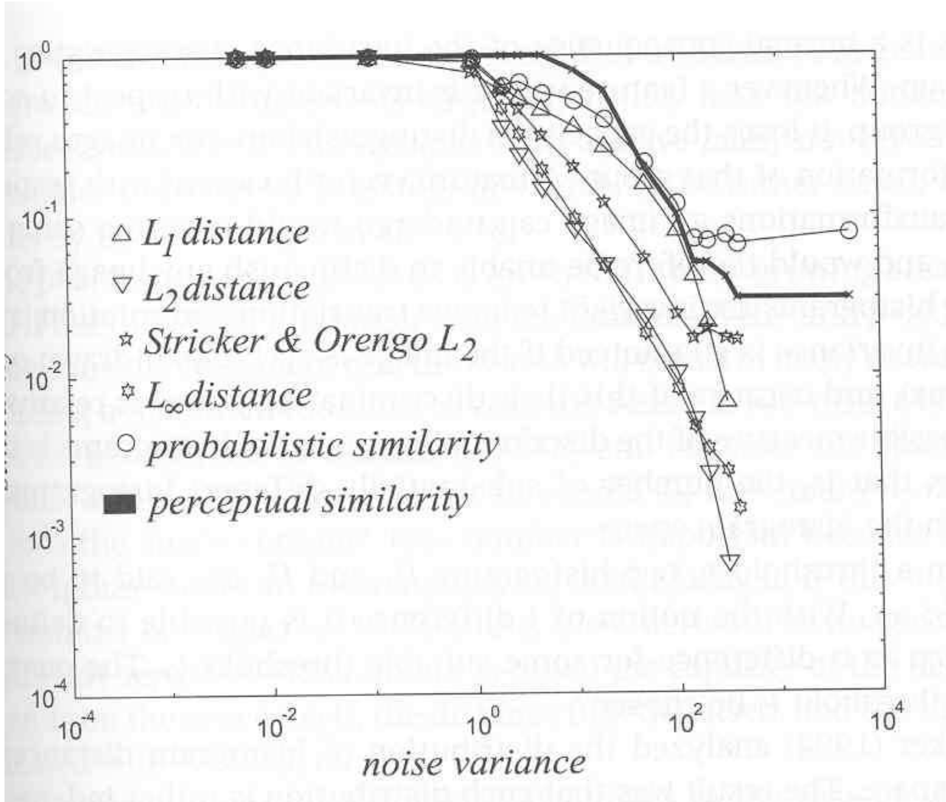
Distance type	d(a,b)	d(a,c)
L_1	2	~0.67
L_2	~0.82	~0.47
L_∞	~0.33	~0.33

None of the distances captures perceptual similarity

Realistic example for problem with distances



Realistic example for problem with distances

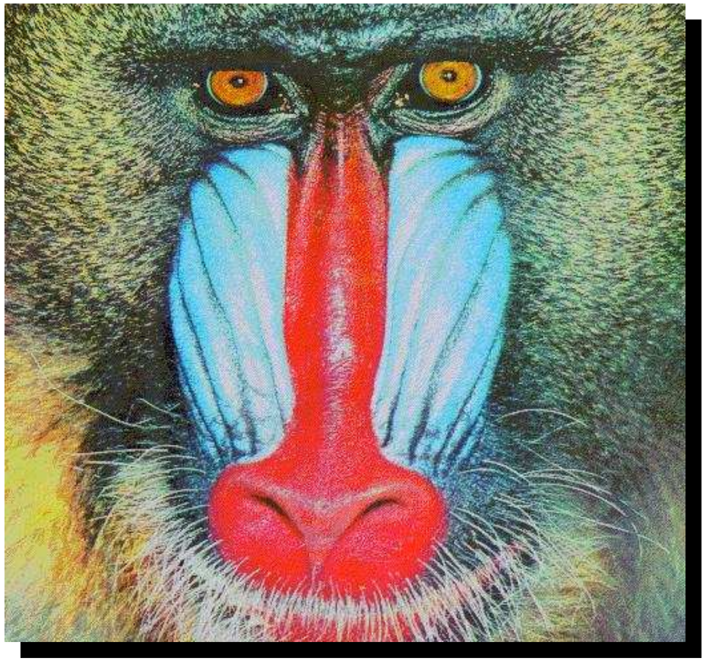


Potential problem with histogram distance

- There are alternative distance measures
- Details beyond the scope of this lecture
- If you seem to have such a problem: look into the literature

Issue: loss of regional information

Partition the image
One histogram per region

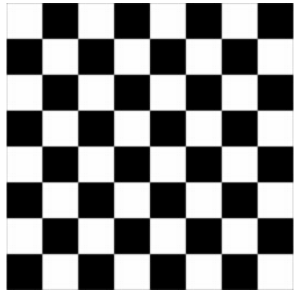


8.2. Texture Features

What´s in the image?

What is texture?

- Texture has no precise definition.
- Texture is a tactile or visual characteristic of a surface.
- Texture primitives (or texture elements, texels) are building blocks of a texture.
- Texel: A small geometric pattern that is repeated frequently on some surface resulting in a texture.

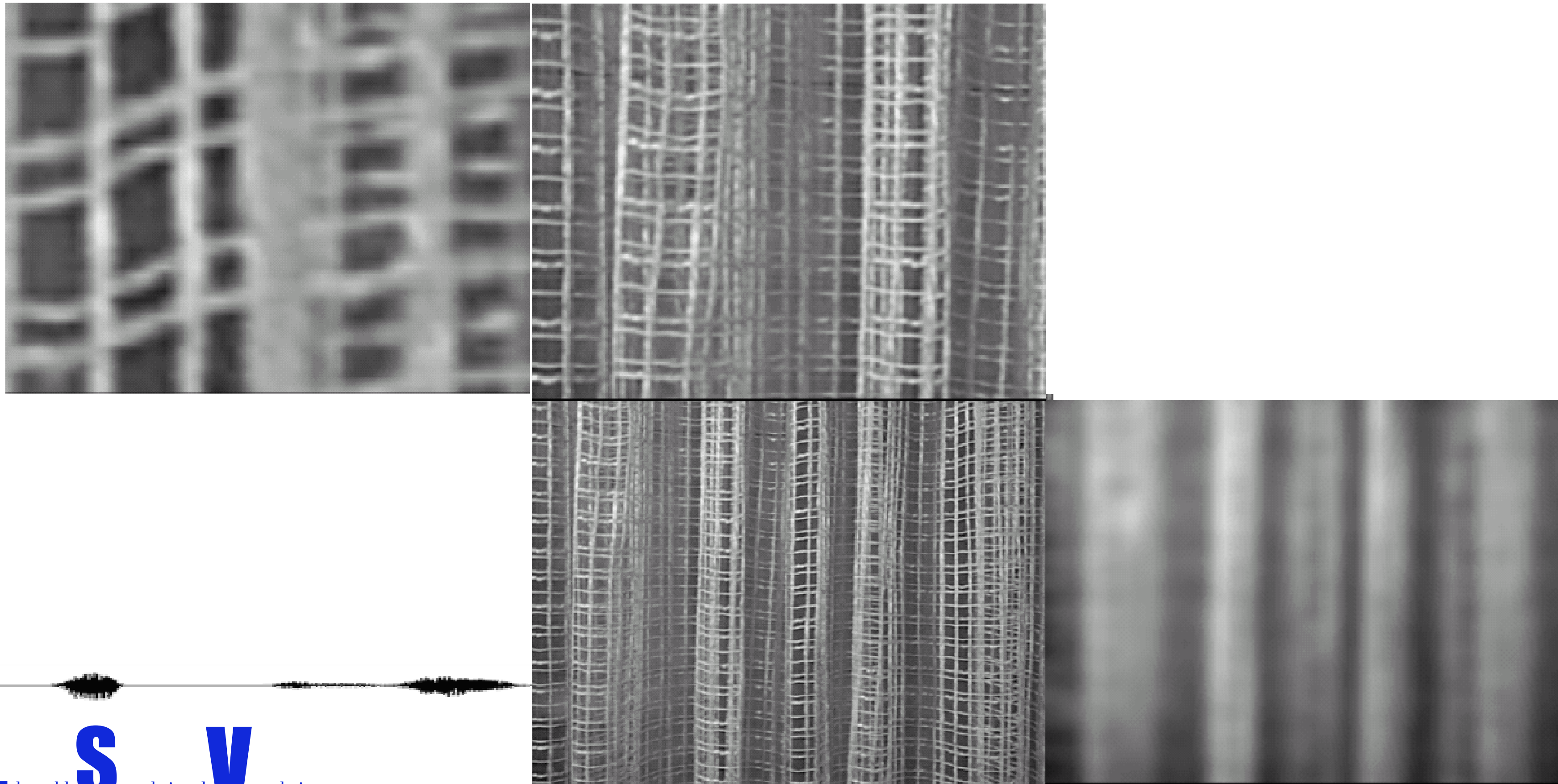


Use of Texture Analysis

- Segment an image into regions with the same texture, i.e. as a complement to gray level or color
- Recognize or classify objects based on their texture
- Find edges in an image, i.e. where the texture changes
- ”shape from texture”
- object detection, compression, synthesis

Difficulties of Texture Analysis

- Which scale to use?



Texture Analysis

- Generic research area of machine vision
- Topic of research for over three decades
- Aim: to find a unique way of representing the underlying characteristics of textures and represent them in some simpler but unique form, so then they can be used to accurately and robustly classify and segment objects.

Types of Texture

- Strong Texture
  - – spatial interactions between primitives are somewhat regular
  - – frequency of occurrence of primitive pairs in some spatial relationship used for description
- Weak Texture
  - – small spatial interactions between primitives
  - – frequencies of primitive types appearing in some neighborhood used for description
- Two basic texture description approaches:
  - – syntactic
  - – statistical

Syntactic texture description

- Not used as widely as statistical approach

- Analogy between texture spatial relationships and structure of a formal language.

- Grammar representation - primitives are terminal symbols, relationships are represented as transformation rules.

First Order Statistics

- Mean $\mu = \sum_{k=1}^{K} k \; p_k$ (hardly a useful feature)

- Variance $\sigma^2 = \sum_{k=1}^{K} (k-\mu)^2 \; p_k$

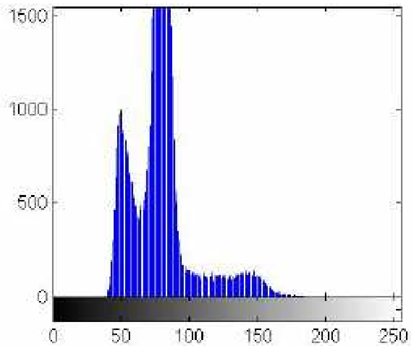
- Skewness $\gamma_3 = \frac{1}{\sigma^3} \sum_{k=1}^{K} (k-\mu)^3 \; p_k$

- Kurtosis $\gamma_4 = \frac{1}{\sigma^4} \sum_{k=1}^{K} (k-\mu)^4 \; p_k - 3$

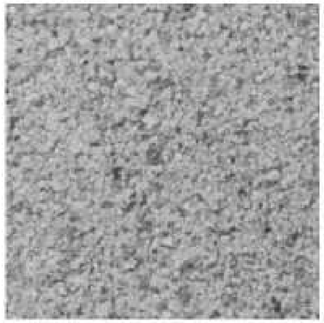
with

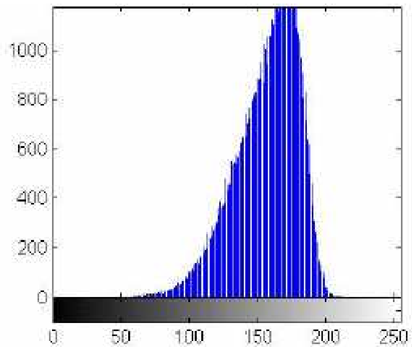
$$p_k = \frac{h_k}{\sum_{k=1}^{K} h_k}$$

Example for first Order Statistics



Brickwall texture
Mean 79.13
Variance 42.73
Skewness 1.37
Kurtosis 5.93





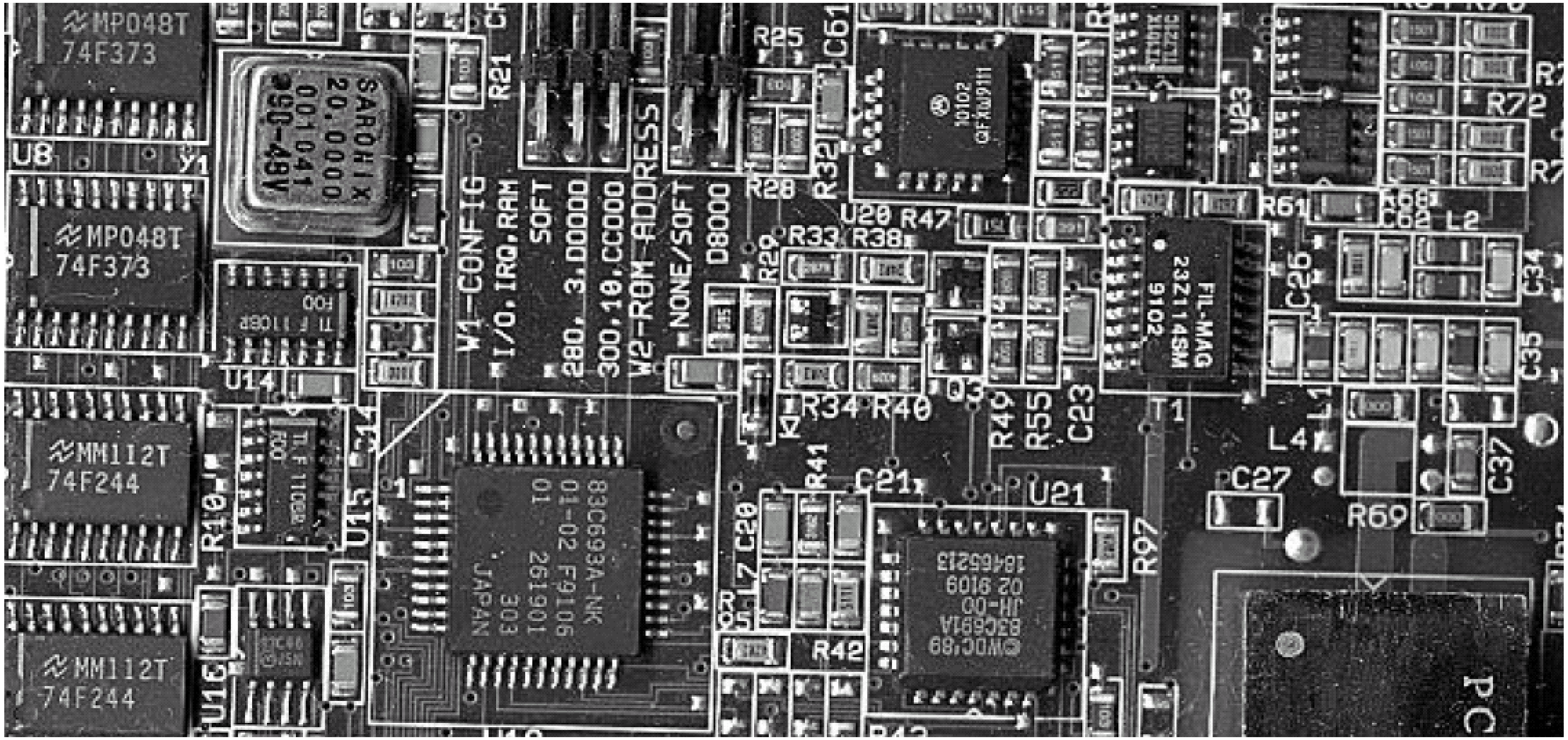
Granite texture
Mean 157.08
Variance 96.9573
Skewness -0.73
Kurtosis 3.25

Autocorrelation Function

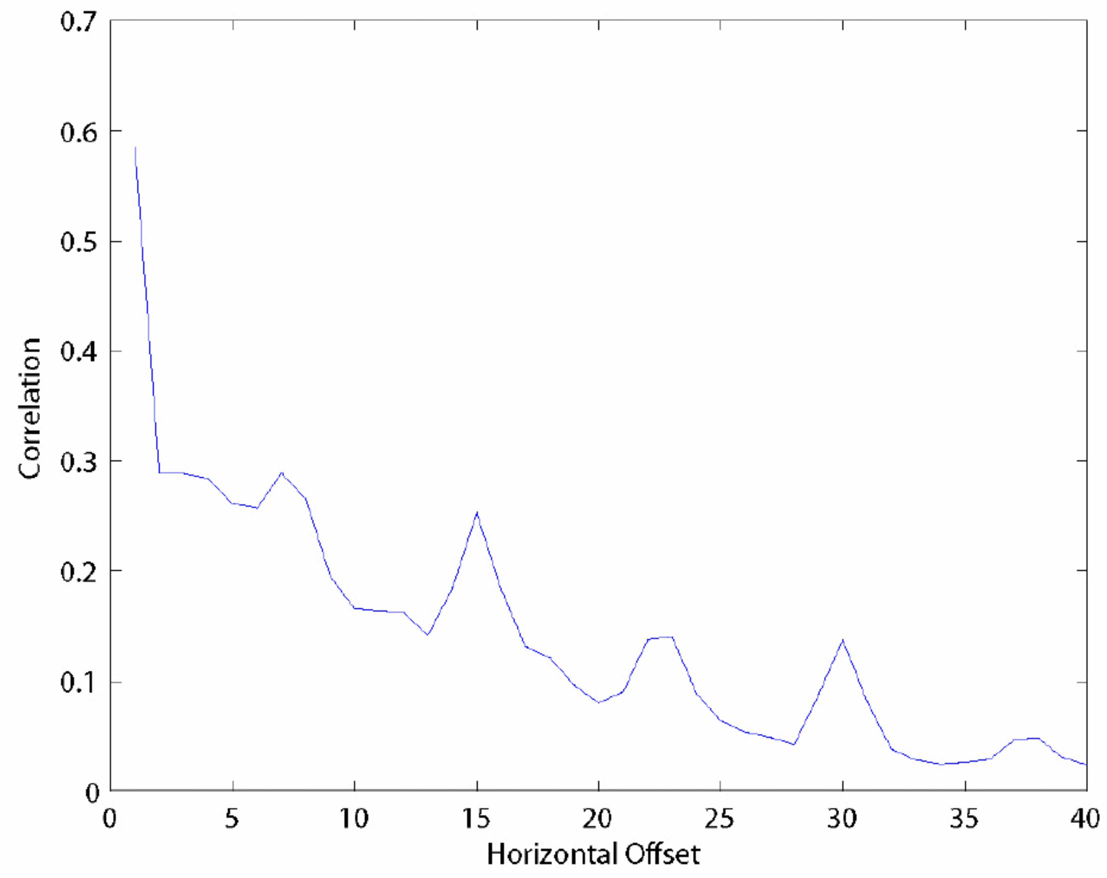
$$\rho_{ff}(i,j) = \sum_x \sum_y f(x,y)\, f(x+i, y+j)$$

- What is the frequency of repetition of structures:
  coarse/fine texture
- How strongly are they correlated:
  Similarity of the texels

Autocorrelation: Example

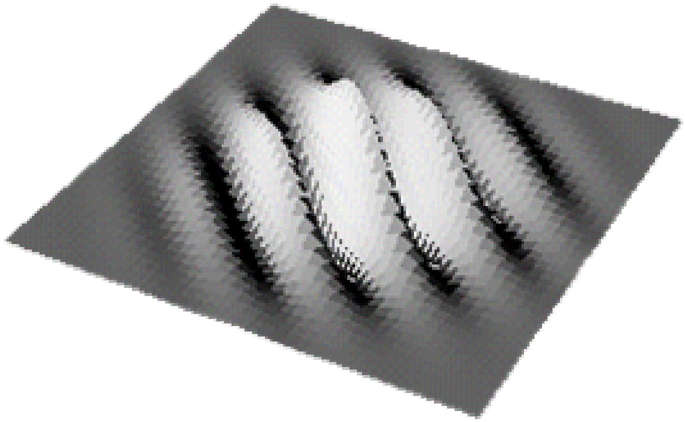


Autocorrelation: Example

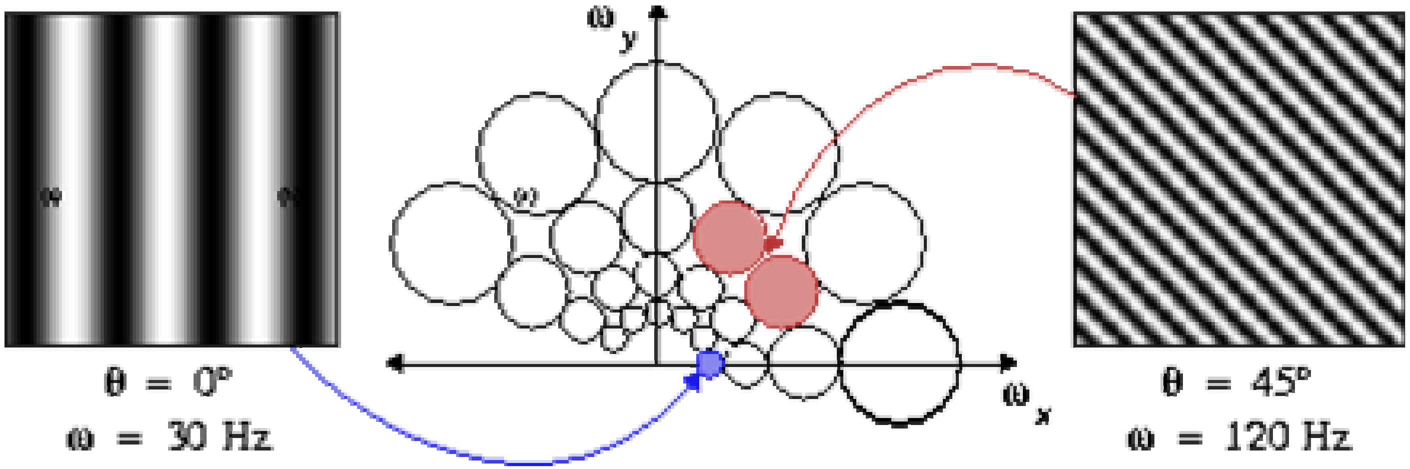


Gabor Filter

- Family of filters
  - Product of Gaussian with traveling waves

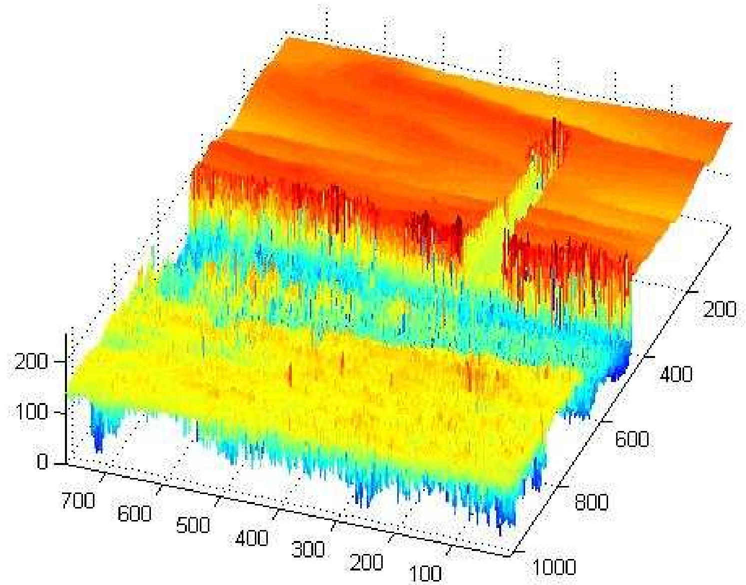


Position of Frequencies for a Set of Gabor Filters



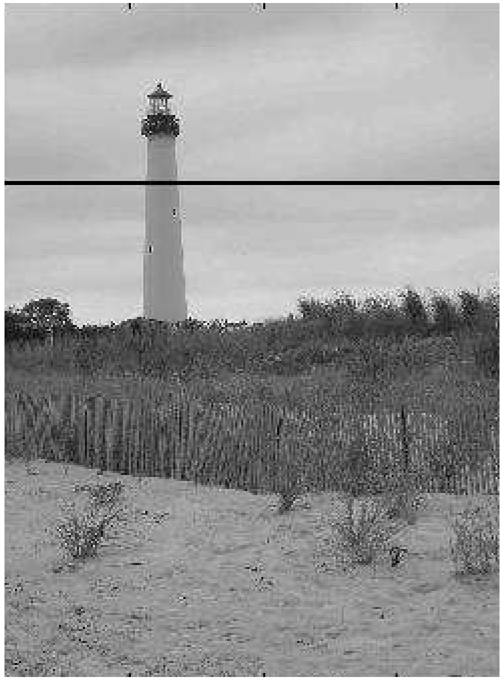
8.3. Edge Information

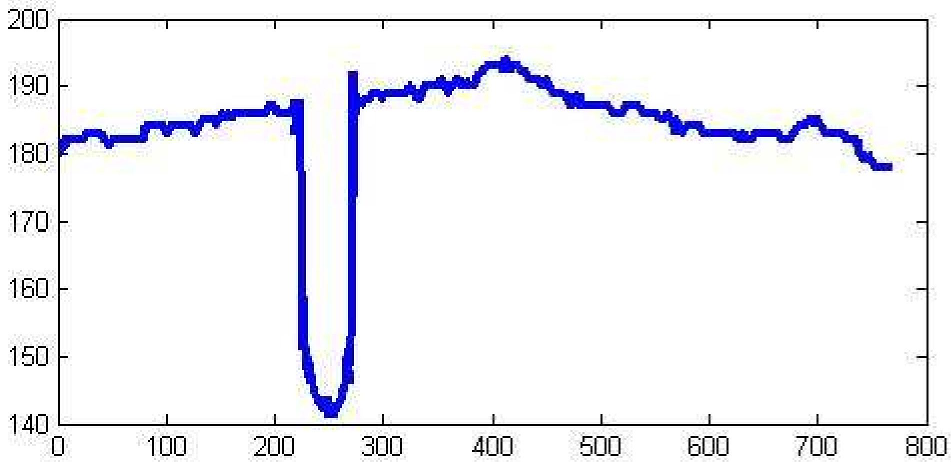
Characterizing Edges



- Images are *discrete* functions indicating the light intensity of a scene
- What happens at an edge?

Characterizing Edges (cont'd)





- Let's look at one line for now

Detecting Edges

- Edges correspond to large discontinuities in the image
- How do we detect such discontinuities?

Gradient Definition

$$\nabla I(x, y) = \frac{\partial I}{\partial x}\hat{x} + \frac{\partial I}{\partial y}\hat{y}$$

- The gradient is a *vector* with magnitude in the *u* and *v* directions equal to the respective partial derivatives
- How do we compute the partial derivative of a discrete function?

Taylor Series…

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \ldots$$

or…

$$f(x-h) = f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) + \ldots$$

- Subtracting the second from the first we obtain

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

Discrete Gradient Estimation

- Discrete functions: use first order approximation of the gradient

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

h corresponds to the step size

- Images: *h* corresponds to the width of 1 pixel =>

$$\frac{\partial I(x,y)}{\partial x} = \frac{I(x+1,y) - I(x-1,y)}{2}$$

$$\frac{\partial I(x,y)}{\partial y} = \frac{I(x,y+1) - I(x,y-1)}{2}$$

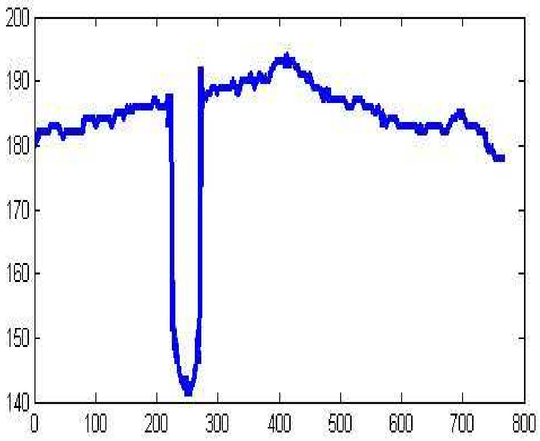
Discrete Gradient as a Linear Filter

- Gradient can be written as a linear filter
- Drop factor 2 because it just scales the image

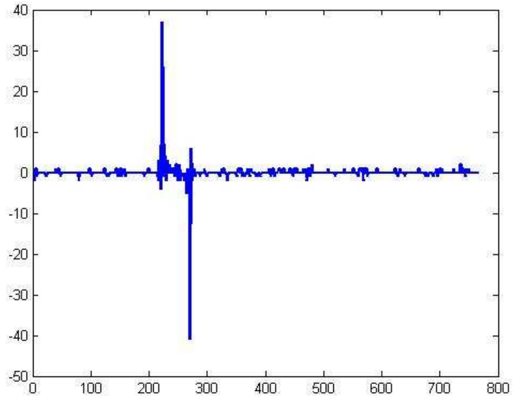
$$\frac{\partial I}{\partial x} = I * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

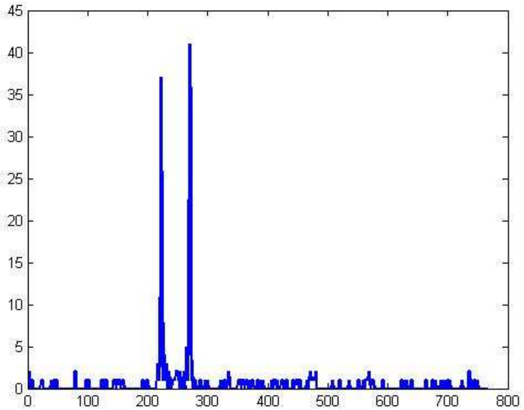
$$\frac{\partial I}{\partial y} = I * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Taking the discrete derivative



$$[-1 \quad 0 \quad 1]$$



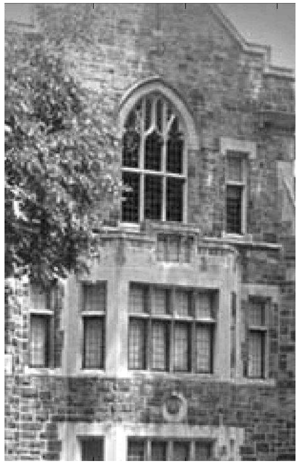


abs()

Basic Edge Detection Step 1

$I(x, y)$

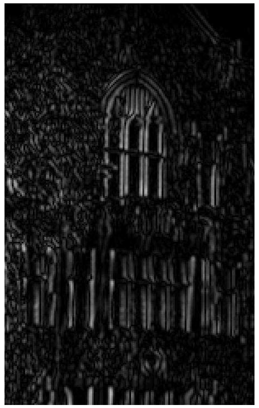
INPUT IMAGE



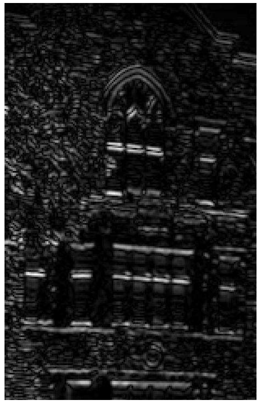
Horizontal
[-1 0 1]

1) Edge Enhancement

Vertical
[-1 0 1]^T



$$\frac{\partial I(x,y)}{\partial x}$$



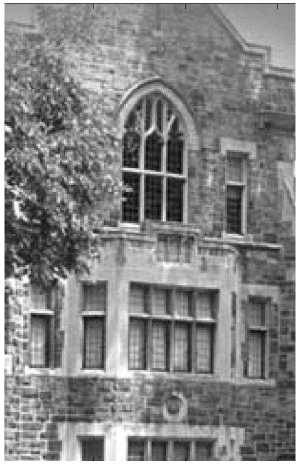
$$\frac{\partial I(x,y)}{\partial y}$$

Issue:
sensitivity
to noise?

Basic Edge Detection Steps 1-2

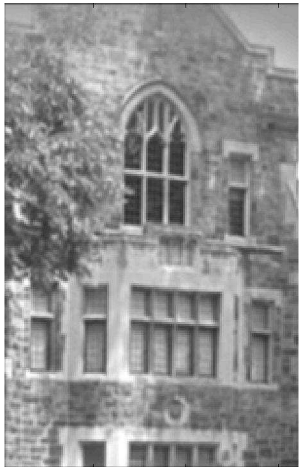
$I(x, y)$

INPUT IMAGE



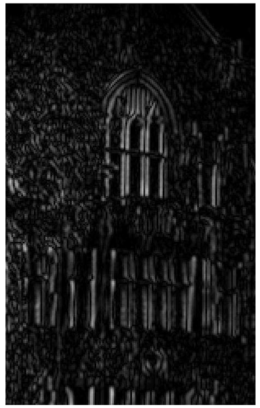
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 16$$

1) Noise Smoothing



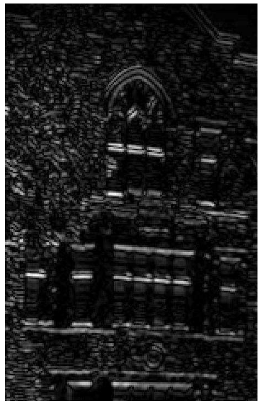
2) Edge Enhancement

Horizontal
[-1 0 1]



$$\frac{\partial I(x, y)}{\partial x}$$

Vertical
[-1 0 1]^T



$$\frac{\partial I(x, y)}{\partial y}$$

Discrete Gradient Estimation

- Gradient is a *vector*
- we have calculated the coefficients in the *x* and *y* directions at each point in the image
- After convolving, we get the magnitude of the gradient from at each point (pixel) from

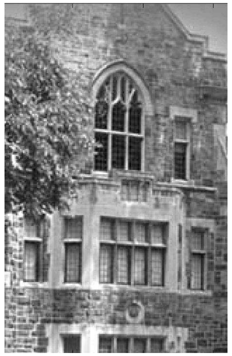
$$G(x,y)=\sqrt{\left(\frac{\partial I}{\partial x}\right)^2+\left(\frac{\partial I}{\partial y}\right)^2}$$

- In practice, we often sum the absolute values of the components for computational efficiency

Basic Edge Detection (cont'd)

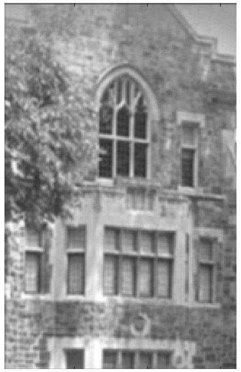
$I(x, y)$

INPUT IMAGE

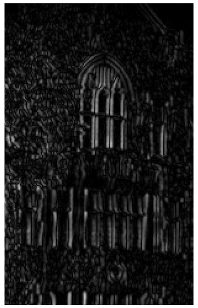


$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 16$$

1) Noise Smoothing



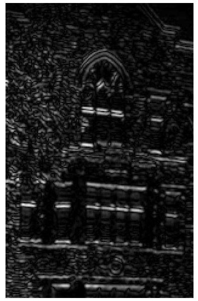
Horizontal
[-1 0 1]



$$\frac{\partial I(x, y)}{\partial x}$$

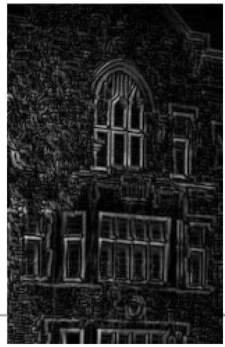
2) Edge Enhancement

Vertical
$[-1\ 0\ 1]^T$



$$\frac{\partial I(x, y)}{\partial y}$$

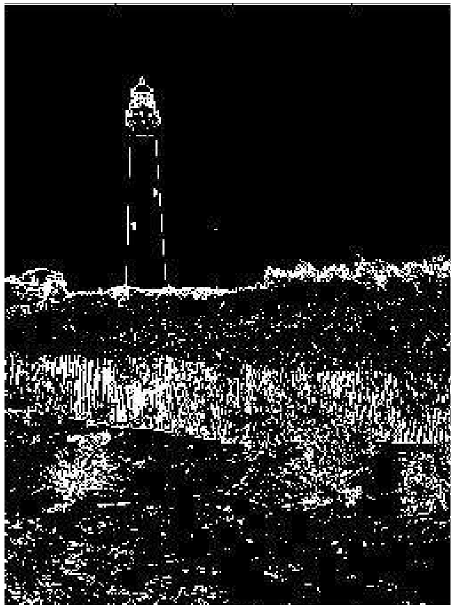
$$|\partial I(x, y)| = \left[\frac{\partial I(x, y)}{\partial x}^2 + \frac{\partial I(x, y)}{\partial y}^2 \right]^{\frac{1}{2}}$$



"GRADIENT" IMAGE

Thresholding

- Remove lighting effects
- Convert to binary image using a threshold

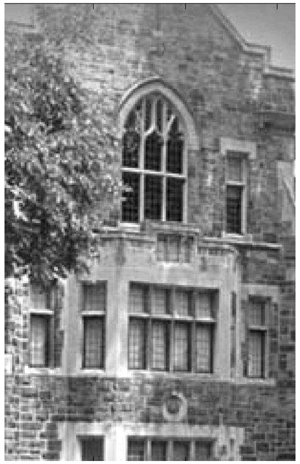


Results from threshold values of 50 and 100

Basic Edge Detection Summary

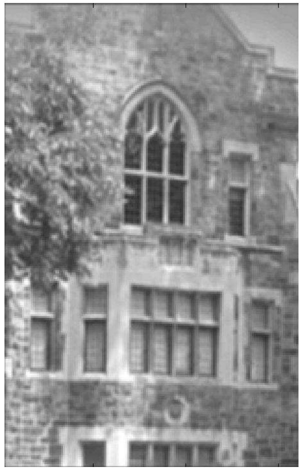
$I(x,y)$

INPUT IMAGE



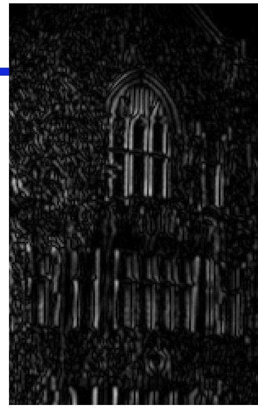
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 16$$

1) Noise Smoothing



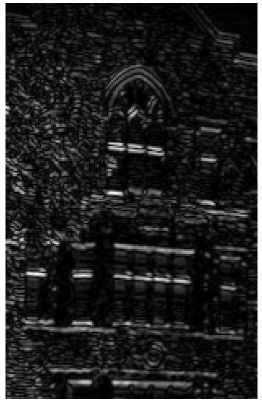
2) Edge Enhancement

Horizontal
[-1 0 1]



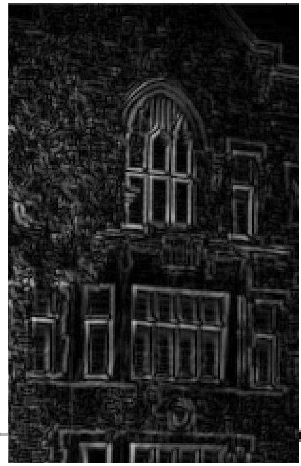
$$\frac{\partial I(x,y)}{\partial x}$$

Vertical
[-1 0 1]^T



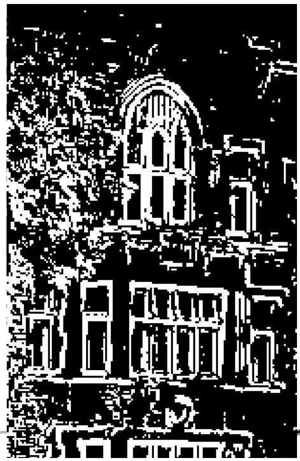
$$\frac{\partial I(x,y)}{\partial y}$$

$$|\partial I(x,y)| = \left[\frac{\partial I(x,y)}{\partial x}^2 + \frac{\partial I(x,y)}{\partial y}^2 \right]^{\frac{1}{2}}$$

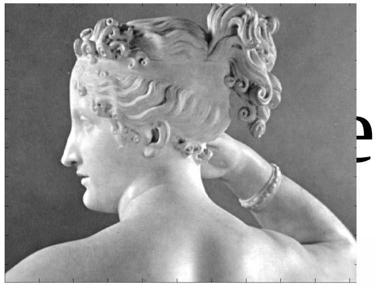


"GRADIENT" IMAGE

3)Threshold



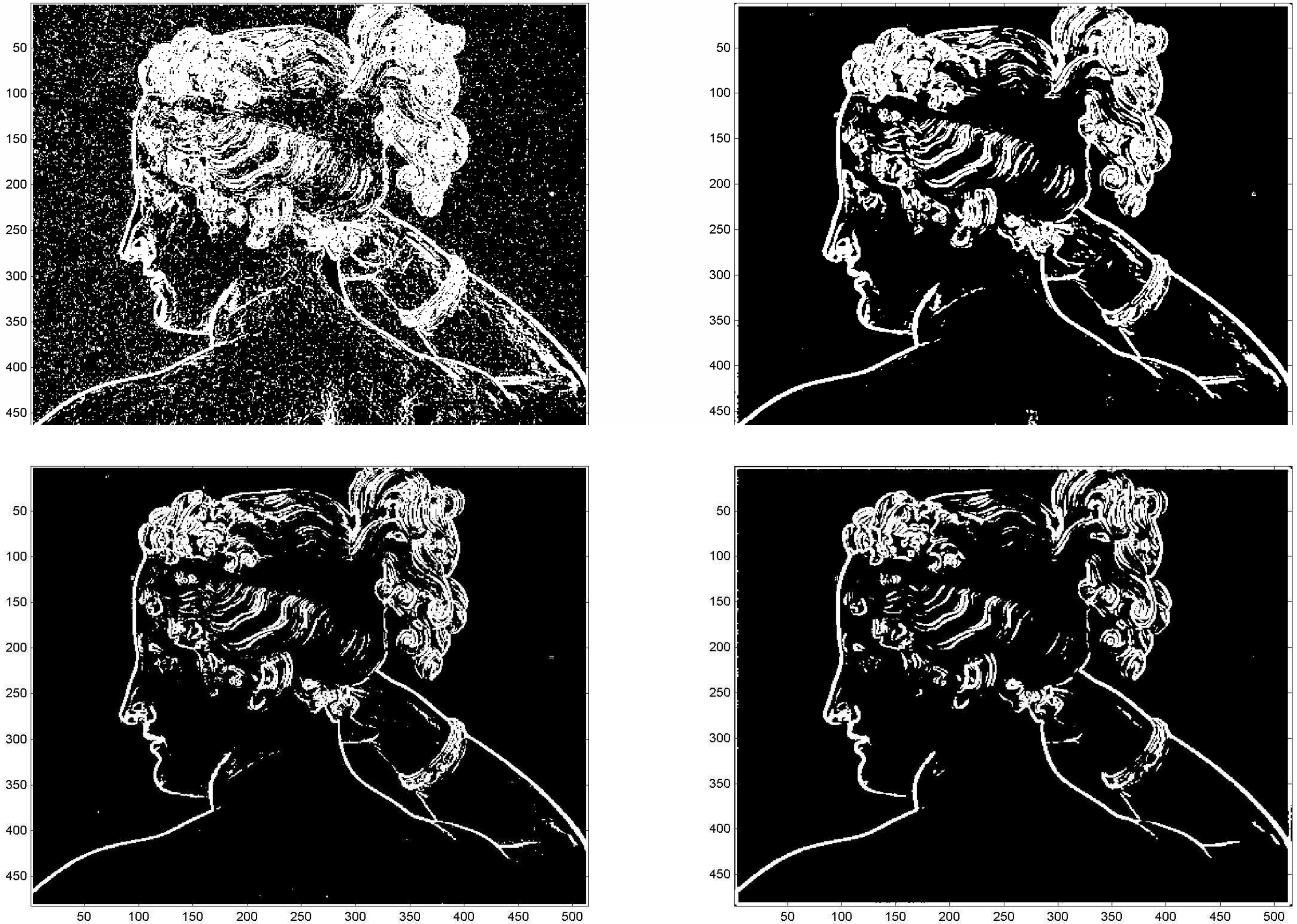
EDGE IMAGE



e effects of Filtering Noise

Threshold 20

Threshold 50



Unsmoothed Edges

Gaussian Smoothing

Lehrstuhl Sprachsignal Verarbeitung

Sobel Edge Detection

- Integrate smoothing and gradient calculation
- Sobel operators: widely spread scheme

$$Sobel_V = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad Sobel_H = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Convolving generates horizontal and vertical gradient images

Other Edge Detectors

- *Prewitt*: similar to the Sobel, but different kernel

$$P_V = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad P_H = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

- *Roberts:* early edge detector kernel

$$R_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad R_2 = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

Summary

Color features:

- use histograms
- issue: robust distances measures

Texture:

- first order statistics
- auto correlation function
- Gabor filter

Summary

- Edges correspond to abrupt changes in image intensity
- Edges can be detected by
  - Smoothing out image noise
  - Estimating the gradient of the image at every point to generate a “gradient” image
  - Thresholding the gradient image